



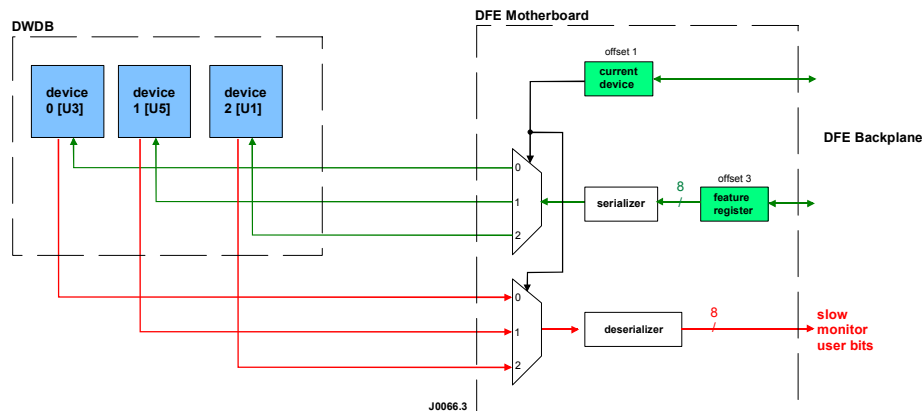
## Jamieson T. Olsen Engineering Note

**Date:** 28 January 2003  
**Rev Date:** 10 February 2003  
**Project:** L1CTTT  
**Doc. No:** 2003-01-28a  
**Subject:** CTTT Control and Status Specification

### Introduction

The L1 CTTT firmware resides in device 0 (U3) of the CTTT “Double Wide” daughterboard. This document describes how to set post-initialization variables and control what status information is passed back to the slow monitor system.

### A New DFE Motherboard – DWDB Interface



This new interface replaces the FIRMWARE REVISION REGISTER logic. A new register called the feature register is located at **base\_addr + 3** on the DFE backplane. This register is readable and writeable from the DFE backplane. When written, the contents of the register are automatically serialized and sent to the current device. (Unlike the older design, a board reset is not needed to initiate this transfer.)

Each DWDB device constantly serializes 8-bits of status information and sends that down to the DFE motherboard. Depending on the value of the current device register, one of the three devices is selected and that data is de-serialized and becomes the slow monitor user bits[7..0].

For this interface to work the DFE Motherboard must have U49 firmware **1.42** or newer. Also this circuitry will not function unless all devices on the DWDB are initialized (ready bit set). In order for a DWDB device to accept bytes from the feature register and also serialize the 8-bits of status information, it must instantiate a module called *mobo\_int.vhd* at its top level.

## New Slow Monitor Status Word

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Board Type				0	MCD	RDY	REG	user bits / current status page							

2003-01-06

DFE Motherboard firmware v1.42 changes the meaning of the slow monitor status word. The master clock detect (MCD) has been moved from bit 0 to bit 10. RDY is set when all of the devices on the CTTT board have been initialized. REG is **cleared** if there is a problem with one or more of the voltage regulators in the DWDB. Bits 7..0 are available for the user bits. The board type for CTTT is 0011.

## Setting Post-Initialization Variables

After the L1 CTOC device is initialized with production firmware, some additional arguments must be passed to it. These are:

- **L3 Pipeline Depth.** This is adjustable from 0-36 crossings deep. The default is 33.
- **Fake L1 record select bits.** For diagnostic purposes, the L1 CTOC can send fake L1 records to the CTTT. Default is zero.
- **Status Page.** The L1 CTTT has a lot of status information, but it can only be viewed in one byte-wide “page” at a time. Default is page 0.

As previously mentioned, the new DFEM-DWDB interface supports writing and reading an 8-bit value to/from each device on the DWDB. This does not provide enough bits to do everything the CTTT needs it to do, so a simple protocol is imposed on this interface. The feature register now has four independent registers, selected by bits 7 and 6:

7	6	5	4	3	2	1	0
0	0		CH	Page Select			
0	1	L3 Pipe Depth (0-36)					
1	0						
1	1					Fake L1	

2003-01-28

So to change the L3 Pipeline depth to 25 do the following:

```
set DFE 3
set device 0
# 25 = 0x19. Now OR 0x19 with 0x40 to set bit 6.
cmd writebyte 3 0x59
```

## Selecting Status Pages

The CTTT supports 16 pages of status information. Each page is a single byte of status information. The current status page becomes the slow monitor user bits for the CTTT board. Previous versions of the status information used logic to stretch error pulses so that spurious errors had a better chance of being sampled by the slow monitor system. However, this method had shortcomings and has been replaced with a status “history” scheme.

Status history works like this: after reset all of the CTTT status pages are set to 0x00. On every 53MHz clock edge the contents of the status pages are updated. If a bit goes high, it stays high until the status page history is cleared. Setting Bit 4 of the Page Register clears the status history for the status pages.

5	4	3	2	1	0
	CH	Page Select			

***NOTE: after powerup or board reset the history bits may contain garbage. So it is good practice to clear the history after these condtions.***

Below are the 16 status pages specific to the CTTT design.

page			S5	S4	S3	S2	S1	S0
0			FX Error	L2	Parity Err	Missing Link	Sync Err	TFW Locked
1								
2								Fake L1 Mode
3			L3 Pipeline Depth [5..0]					
4								
5	0	1	0	1	0	1	0	1
6	1	0	1	0	1	0	1	0
7								
8				Miss-link4	Miss-link3	Miss-link2	Miss-link1	Miss-link0
9						Miss-link7	Miss-link6	Miss-link5
A				Sync-err4	Sync-err3	Sync-err2	Sync-err1	Sync-err0
B						Sync-err7	Sync-err6	Sync-err5
C				Parity-err4	Parity-err3	Parity-err2	Parity-err1	Parity-err0
D						Parity-err7	Parity-err6	Parity-err5
E				Patt-err4	Patt-err3	Patt-err2	Patt-err1	Patt-err0
F						Patt-err7	Patt-err6	Patt-err5

2003-01-28a

Status Page 0 is kind of an overview page – it provides the user a summary of the error conditions that the CTTT device is seeing:

**Parity Err.** logical OR of the eight Parity Error bits in pages 12 and 13.

**Missing Link:** logical OR of the eight Missing Link bits in pages 8 and 9.

**Sync Err.** logical OR of the eight SYNC\_ERR bits in pages 10 and 11.

**TFW Locked.** The CTTT extracts two trigger framework bits (First Crossing, CFT\_RESET) from the CTOC L1 records. From these control bits it can determine the beam timing and it maintains a tick and turn counter. In the process of calculating the tick and turn numbers, the CTTT knows when to *expect* the First Crossing (FX) control bit. If the FX bit comes at the expected time, then the tick and turn counter remains LOCKED. If the FX bit comes at an unexpected time, the counter module zeros the tick and turn counts and clears this bit. If the CTTT loses LOCK it needs to see a CFT\_RESET followed by a FX bit in order to relock.

**L2.** Set when a L2 record comes into the CTTT on link2. Cleared by the Clear History control.

**FX Error:** Every time a L1 record comes into the CTTT all of the FX bits are compared. This error bit is the logical XOR of the individual FX bits— if the FX bits disagree, this bit is set and stays set until the history is cleared.

**Status Pages 1-3** always returns the value of their associated registers. They are not affected by the status history control.

**Status Pages 5-6.** 0x55 and 0xAA, respectively. Used for debugging the interface.

**Status Pages 8 and 9.** The Missing Link error bit for each input link is listed here. If a link has not seen a Beginning of Record (BOR) transition in at the last 63 clock cycles, the corresponding bit is set and stays set until the history is cleared.

**Status Pages 10, 11.** After the links have been synchronized, a circuit checks that each link's BOR signal is aligned with the master BOR signal. If there is a synchronization problem with a link, the corresponding bit is set and stays set until the history is cleared. Only considers non-missing input links.

**Status Pages 12 and 13.** As L1 records come into the CTTT, the horizontal parity bits are checked for frames 1-6. If there is a parity error in the first 6 frames of the L1 record then the corresponding bit will be set and stays set until the history is cleared. Only considers non-missing input links.

**Status Pages 14 and 15.** The CTTT contains test record detection circuitry. The L1CTOC can be placed into a special mode where it sends a L1 test record to the CTTT. *Used only for CTOC to CTTT LVDS link tests.*

Some Examples:

To change the L3 pipeline depth to 30 and confirm it:

```
set DFE 3
set device 0
# change the L3 pipe depth to 30. (0x1E || 0x40 = 0x5E)
cmd writebyte 3 0x5E
# now set the current page to 3
cmd writebyte 3 0x03
# now the user bits = status page 3
get userbits 3
# user bits should equal 0x1E
```

To clear the history and then rapidly collect all of the status pages:

```
set DFE 3
set device 0

# clear history and set current page to 0
cmd writebyte 3 0x10
# read status page 0
get userbits 3

cmd writebyte 3 0x01
get userbits 3

cmd writebyte 3 0x02
get userbits 3

cmd writebyte 3 0x03
get userbits 3

....
```